
.....

Corvus PCI

high resolution positioning controller

Manual

- About this documentation** 4
 - Symbols in this documentation 5

- Chapter -1- Introduction** 6
 - Presentation of the controller** 7
 - Function survey** 9

- Chapter -2- Technical data** 10
 - Specifications** 11
 - Layout and Dimensions** 13

- Chapter -3- Installation** 14
 - Safety notice** 15
 - Hardware installation** 16
 - Software driver installation** 18
 - Driver Installation Procedure 18
 - Installing Windows-98 Drivers for the Nm9845 18
 - Removing Previously Installed Drivers 21
 - Using NMUNINST.EXE to remove old Drivers 21
 - Removing the old Drivers manually 22
 - Standard Power Supply connection** 23
 - Setting-up the RS-232 connection** 24
 - COM Port settings of Corvus PCI 24
 - Fixed Covus PCI COM Port settings 24
 - Corvus PCI Baudrate selection with Jumper array JA5 25
 - JA5 Baudrate selection 25
 - Testing communication 26
 - Testing with WinPos 26
 - Motor adaptation** 27
 - Axes designation 27
 - Motor types 27
 - Motor characteristics 27
 - Motor configuration command 28
 - Measures against system resonances 29
 - Motor cable 30
 - Motor wiring 31
 - Stepper motor with 2-Phases 31
 - Stepper motor with 3-Phases 31
 - Limit Switch Configuration** 32

- Limit Switch Functions 32
- Limit Switch Type adaptation..... 33
- Adaptation to PNP or NPN Types with Jumper Array JA7 33
- JA7 Function..... 34
- JA7 configuration examples: 34
- Setting the Switch power supply with Jumper Array JA8 35
- JA8 Function..... 35
- JA8 configuration example 36
- Limit Switch adaptation to Opener or Closer Type 37
- Limit Switch wiring 38

- Chapter -4- Functions** 39
 - Limit Switch movement** 40
 - Programmed Limit Switch movement in detail 41
 - Limit Switch movement to the cal-switch 42
 - Limit Switch movement to the rm-switch 42
 - Limit Switch movement with the joystick** 43
 - Limit Switch movement with the joystick in detail 43
 - Working range and coordinate system 43
 - Notices for the Limit Switch movement 44
 - Programmed positioning** 45
 - Venus-1 syntax and examples 46
 - Axes interpolation 47
 - Velocity and acceleration..... 47
 - Velocity profile of a linear acceleration function 48
 - Movement profile of short distances..... 48
 - Movement profile of a sin²-acceleration function 49
 - Closed-loop function (Option)** 50
 - Venus-1 closed loop commands: 50
 - Supported measuring systems** 51
 - A/B Quadrature input..... 51
 - SIN/COS Module (Option) 52
 - SIN/COS Status LED..... 52

- Chapter -5- Electrical Interface** 53
 - Electrical Interface Locations and Designation** 54
 - Power connector CN16** 55
 - Motor and Limit Switch Connectors CN1-CN3** 56

About this documentation

This manual provides detailed* information on the hardware and Software features of the Corvus PCI Controller. This information includes features, specifications and connector description.

*In Manual Version 1.0 only the Standard functions are described in detail.

In addition, this guide takes you through the process of installing your board and also describes the installation of the delivered package for the operating system Win98.

Symbols in this documentation

To clarify the content following symbols are used.





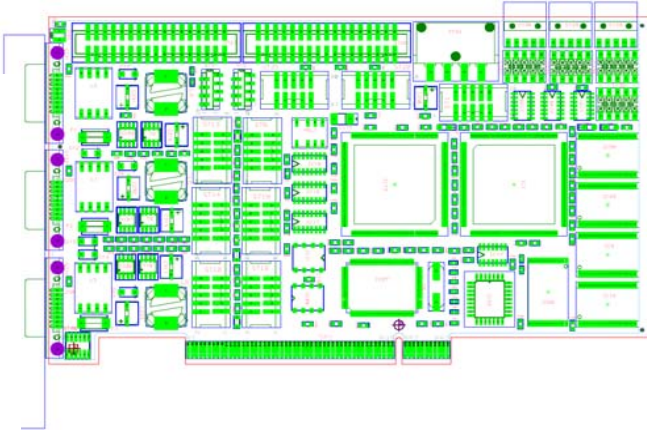
Symbol	Description
	Warning. The informations must be observed strictly
	Hint.
	This function can be enabled with a release code.
	This function must be installed from the factory personal or experts.
Venus-1	Venus-1 command

Table 1. Symbols

Chapter 1

Introduction

Presentation of the controller



Technology

Corvus PCI is a complete stepping motor controller for alternatively 2 or 3 axes and Motor torque up to 1.5 Nm. Controller and motor drivers are integrated onto a single PCI board.

Corvus PCI uses a 32-bit, 133MHz Risc processor working with a Real Time Operating System (RTOS) for scalable performance and high precision synchronized control.

A digital PID servo loop updates all axes with a cycle time of 250 μ s for precise velocity profile tracking and accurate positioning.

A extremely micro-step resolution up to 600.000 steps/rev. provides ultra-smooth stepper positioning capability for high precision applications.

Low impedance MosFet motor drivers guarantee high motor performance with less power dissipation at the board.

Motion

Corvus PCI provides point-to-point, jogging, linear or interpolation moving including velocity controlled moves with the possibility of on the fly velocity updates.

Software limits can be set to improve systems safety. An advanced origin search routine includes encoder index pulse consideration for precision homing.

Resolution, velocity, acceleration

The elaborated controller technology enables a low-noise positioning and at the same time an extremely high stepping resolution. In the programmable mode, a positioning resolution of 1,5 nm is achieved.

The resolution in the joystick mode is 23.2 fm (femtometer). Due to the high computing power Corvus PCI has a high speed and acceleration dynamic from 15nm/s until 25 rev/s. Linear- or \sin^2 -acceleration is also supported.

Options

Corvus PCI is supplied with a great variety of options. See table 2 page 9.

Communication

For communication a PCI I/O bridge is used, based on a Nm9845 Netmos chipset.

With this concept the installation of the Corvus card is trouble free. Drivers for the most popular operation systems are available.

Programming

Corvus PCI is programmed with ASCII command language Venus-1. With more than 100 commands. Motion macro execution is also provided.

For easy motion programming the controller is working with with the physical units μm , mm, inch, m, mm/s, mm/s²

Manual Operation

Manual operation via a Joystick is possible.

Firmware Update

To keep the Corvus PCI firmware always up to date the firmware can be easily updated without removing the card.

Function survey

The following list summarizes the main features of the Corvus PCI Motion controller/driver. Standard and optional features are remarked.

Features	Standard or Option
• 2 axes	• Standard
• 3 axes (additional 3 rd. axis)	• Option / Code
• Max. motor phase current: 1.5A	• Standard
• Max. motor driver resolution: 1,5nm	• Standard
• Motor revolution (200 step motor): 15 rev/s	• Standard
• Motor revolution (200 step motor): 25 rev/s	• Option / Hardware
• Power Supply 12V, internal from PC	• Standard
• Power Supply 24V/75W, external supply	• Option/Hardware
• Joystick operation	• Option/Hardware
• ASCII command language	• Standard
• Motion macro execution	• Standard
• Error correction 1D, open loop and closed loop	• Standard
• Sin ² - acceleration (S-Curve)	• Standard
• Closed Loop (RS422 digital position encoder inputs)	• Option / Code
• Closed Loop (Sin/Cos 1Vss inputs)	• Option / Hardware
• Home switch, pos. / neg. limit switches interface	• Standard
• Digital I/O, 3 inputs / 3 outputs	• Option / Code
• Digital I/O, 8 inputs / 8 output	• Option / Hardware
• Analog outputs, 2 outputs, 0-1V, 8Bit	• Option / Hardware
• Trigger Out Function, 4 Khz (cycle =250µs)	• Option/ Hardware
• Motor enable/disable input	• Option / Hardware
• Firmware update	• Standard
• Communication PCI to RS-232 bridge	• Standard
• CAN-Bus provided	• Option / Hardware
• Compact, half size PCB	• Standard

Table 2. Function survey

Standard: Basic configuration

Option / Code: Hardware is prepared, release code is necessary

Option / Hardware: Additional hardware extension with PC bracket

.....

Chapter 2

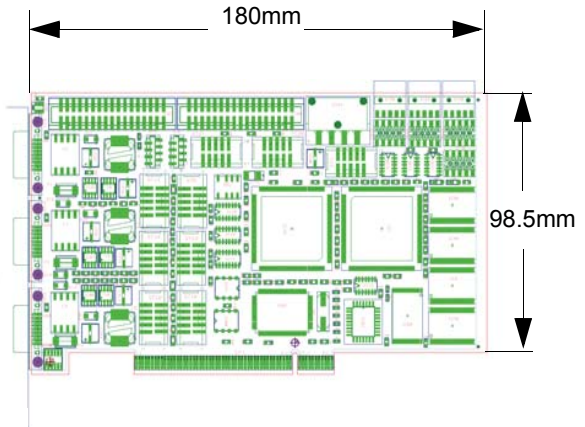
Technical data

Specifications

Number of axes	<ul style="list-style-type: none"> • 2 or 3
Phase current	<ul style="list-style-type: none"> • 0.2A...1,5A
Velocity max. Velocity min.	<ul style="list-style-type: none"> • 25 rev/s • 15.26 nm/s
Acceleration max. Acceleration min.	<ul style="list-style-type: none"> • 1m/s² • 15.26 nm/s²
Supported motors	<ul style="list-style-type: none"> • Stepper motor, 2-Phases or 3-Phases, bipolar,
Programming	<ul style="list-style-type: none"> • ASCII commands Venus-1 • Windows user interface • Software library (dll)
Resolution in programming mode	<ul style="list-style-type: none"> • 1.526 nm
Resolution in Joystick mode	<ul style="list-style-type: none"> • 0.233 nm (233 femto meter)
Positioning modes	<ul style="list-style-type: none"> • Absolute, relative, vector, error correction • speed controlled with on the fly velocity updates
Acceleration modes	<ul style="list-style-type: none"> • linear (trapezoid), sin² (S-Curve)
Closed Loop (Option)	<ul style="list-style-type: none"> • Sin-Cos 1Vss or RS422 quadrature
Processor	<ul style="list-style-type: none"> • 32 Bit Risc Processor
Operating system	<ul style="list-style-type: none"> • Real Time Operating System (RTOS)
Motor driver	<ul style="list-style-type: none"> • Digital, MOS-FET technology
Communication	<ul style="list-style-type: none"> • PCI to RS-232 Com Bridge via Nm9845 Chip set • Data rate up to 115200 Baud
Limits	<ul style="list-style-type: none"> • 2 x Limit Switch inputs for each axis • (isolated)

Input/Outputs (Option)	<ul style="list-style-type: none"> • 3 x Digital inputs • 3 x Digital outputs • 2 x Analog outputs 0-1V, 8 Bit • 3 x Temperature sensor inputs (-25°C - 100°C)
User Output power	<ul style="list-style-type: none"> • +5V, 1000 mA • +12V, 300 mA
Increm. Digital Encoder (Option)	<ul style="list-style-type: none"> • 3 Channel differential, incremental, RS422 • Multiplication factor: x4 • Encoder index pulse consideration • Input frequency 20 Mhz.
Sin-Cos Analog Encoder (Option)	<ul style="list-style-type: none"> • 3 Channel differential, incremental 1Vptp. • Multiplication factor: x12,865 or x205,884 • Encoder index pulse consideration • Input frequency 500,000 sine/periods/second
Safety functions (Option)	<ul style="list-style-type: none"> • Motor disable interface • Isolated relay contacts
Accessories	<ul style="list-style-type: none"> • Analog Joystick interface with 2 or 3 axes • Analog Joystick with graphic display • Digital Encoder Handwheel
Trigger Out Function (Option)	<ul style="list-style-type: none"> • Position Synchronized Output (PSO) • Update 4khz
Dimensions	<ul style="list-style-type: none"> • L x W: 180 x 98.5 (mm)
Power motor driver	<ul style="list-style-type: none"> • +12V DC +/-5%, 2,5A max. Connector: 4pin Standard Molex, female Supplied from internal PC Power Supply
Power motor driver external (Option)	<ul style="list-style-type: none"> • +24V DC +/-5%, 2,5A max. Supplied via an extended PC-Bracket and external Power Supply
Standards	<ul style="list-style-type: none"> •

Layout and Dimensions



Chapter 3

Installation

Safety notice



The controller is developed, produced, checked and documented in consideration of the relevant standards. If it is used according to regulations, there are no dangers for persons and things. Use according to regulations implies that the device is solely used in the way that is described in here and that the stated safety advices are followed.



Hardware installation

Step 1

Power down your computer and disconnect the power cord. Be sure to remove the power cord completely from your computer.

Step 2

Open the computer box:

Remove the screws that hold the panel/shell of your computer (located on the back of the computer). Be sure to put the screws removed in a safe place, as you will need them again when replacing the shell on your computer.

Step 3

With the power cable disconnected from your computer, touch the metal part of the case to ground yourself. This will discharge any static electricity from your body.

Step 4

Locate an available PCI expansion slot. This will be in the area of your computer where other cards that look similar to the Corvus PCI card are installed.

Step 5

Visually confirm that the Corvus PCI Card will fit into the available slot you have chosen.

Step 6

If necessary, remove the knockout on the back of your computer which corresponds to the PCI slot you have confirmed as your selection. If there is a screw, place it in a safe place as you will be using it to attach the Corvus PCI Card.

Step 7

Push the Corvus PCI Card firmly into the PCI slot. Apply pressure as needed (but do not force the card) until the edge connector is completely in the slot.

Step 8

Connect the molex power supply to the Corvus PCI Card power connector CN16.

Step 9

Mount the Corvus PCI Card in place with a screw. Make sure the screw is firmly tightened so the Corvus PCI Card does not slide out of its seated position.

Step 10

Replace the computer's panel/shell and screw it back in place with the screws you removed in Step2

Step 11

Now that your Corvus PCI Card is installed and the shell is in place, you can safely connect the power cable and start with the software installation.

For Software driver installation go through following description.

Software driver installation

Driver Installation Procedure

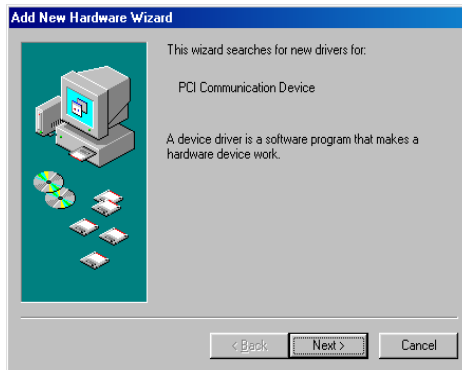
This document will present the steps needed to install or update the Drivers Corvus PCI for Netmos Nm9845 PCI to Dual Serial.

The actual sequence of events is different for each Operating System. This example will describe installing the Driver under Windows-98.

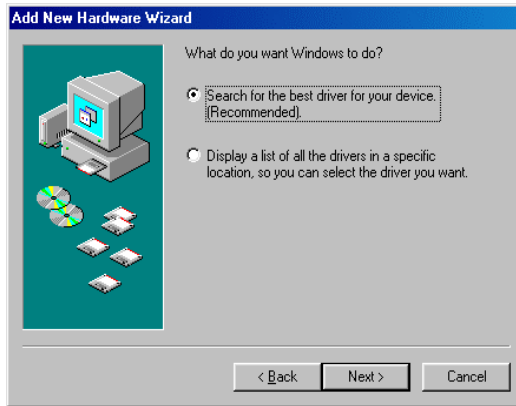
Installing Windows-98 Drivers for the Nm9845

- Shutdown Windows
- Turn the Power to the computer OFF.
- Insert Corvus PCI into the computer.
- Re-boot Windows
- Windows should detect the new card, and automatically invoke the “Add New Hardware” wizard. If not, manually start the wizard.
- Open the Control Panel
- Open “Add New Hardware”

i



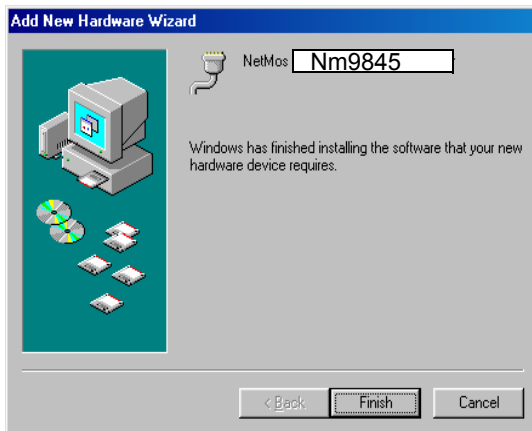
- Insure that the product listed is at least similar to the Drivers you downloaded. The Nm9845 is actually listed as: “PCI Communication Device”
- Click “Next” to start the process.



- Select “Search for the best driver for your device. (Recommended)”
- Click “Next” to proceed.



- Browse to the directory on the Hard Disk containing the Corvus PCI files.
- Click “Next” to continue.



- Click “Finish” when Windows says it has finished installing the Drivers.
- Windows should display additional dialog boxes saying it has found new hardware, and is installing the appropriate software to handle the new devices. These steps should proceed without any additional input from you.

Removing Previously Installed Drivers

Sometimes it will be desirable or even necessary to remove the previously installed drivers, so that a new, clean installation can be performed. Drivers can get corrupted when being written to their installation medium (CD-ROM etc.). If this happens, the drivers may not install properly. Even if the installation proceeds without errors, the drivers may not work. Sometimes older outdated drivers are shipped with a product. Newer machines may require newer updated drivers, but the older drivers prevent them from being installed.

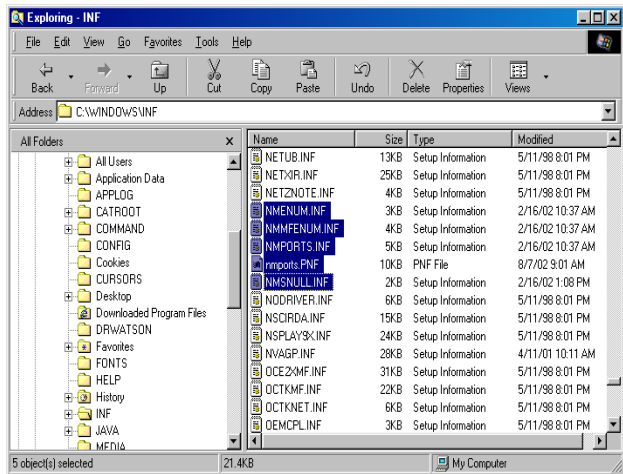
The easiest way to remove old drivers is to use the NMUNINST.EXE program. If this is not possible, it might be necessary to remove the old drivers manually.

Using NMUNINST.EXE to remove old Drivers

- Turn the computer OFF.
- Remove the peripheral card. This will insure that none of the old Driver components will be used, allowing them to be removed from the system.
- Boot the system up normally.
- Locate and run the NMUNINST.EXE program. This program is usually loaded into the “windows\system” or “windows\system32” directory. Using the “find” command from the “Start Menu” will help locate this file.
- Re-boot the computer with the card still removed from the system.
- Shut the computer OFF. Re-install the card, and turn the computer back ON.
- When installing the Drivers this time, use the ones downloaded from the MosChip web-site. It is likely that the reason the old drivers needed to be removed was because they were attempting to use files belonging to a different version of Windows from the one currently running on the machine (parts of Windows-98 loaded instead of Windows-NT for example).

Removing the old Drivers manually

- Turn the computer OFF.
- Remove the peripheral card. This will insure that none of the old Driver components will be used, allowing them to be removed from the system.
- Boot the system up normally.
- Create a temporary directory that will hold the old driver files until we are sure they can be deleted.
- Move anything that looks like it might be a NetMos Driver file from the “windows\inf” directory to the temp directory. These will be files that start with the letters “nm”.

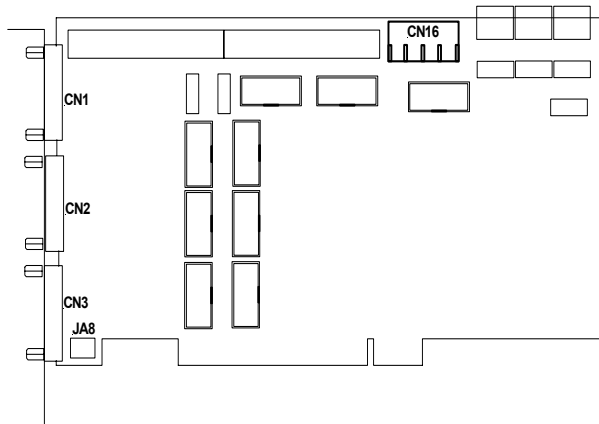
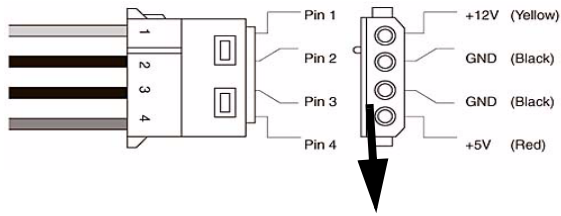


Standard Power Supply connection

The following diagram shows the power supply system of the Corvus PCI.

The motor power +12V and controller power +5V* must be provided on connector CN16 with a standard 4 Pin Molex connector.

*+5V is also applied to Pin A5 at the PCI connector.

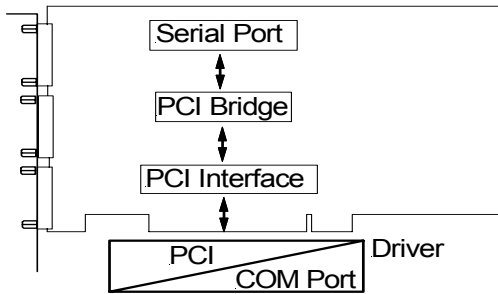


Setting-up the RS-232 connection

Corvus PCI is equipped with a PCI to COM Port bridge to communicate with the host (see drawing).

With the appropriate driver the programming interface is a COM Port.

Therefore the correct COM Port settings are necessary.



COM Port settings of Corvus PCI

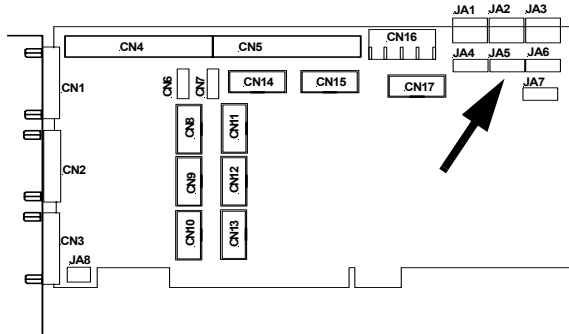
For proper operation, the COM Port settings of Corvus PCI and the Host COM Port have to correspond accurately.

The following settings of Corvus PCI are fixed and unchangeable.

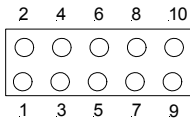
Fixed Covus PCI COM Port settings

• Data bits	• 8
• Stop bits	• 1
• Parity	• no
• Handshake	• no

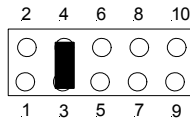
Corvus PCI Baudrate selection with Jumper array JA5



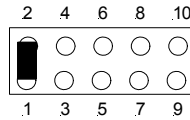
JA5 Baudrate selection



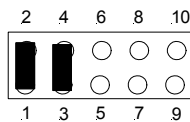
9600 Baud



19200 Baud



57600 Baud
(factory setting)



115200 Baud

Testing communication

You have several choices for sending test commands to Corvus PCI:

You can enter commands from WinPos or you can write your own program. Command language is named Venus-1.

The function of these commands is described in the Manual Venus-1 Command Language.

Testing with WinPos

WinPos LT is added for free to each controller and is qualified to test all functions of the controller.

The most important configurations in WinPos:

- **Connecting the controller with the PC**
- **Configuring host mode**
- **Switching on the controller**
- **Starting WinPos**
- **Configuring number of COM Port and Baudrate**
- **Execute "connecting" WinPos**
- **Opening Venus command line and testing commands**

Motor adaptation

Axes designation

The axes are named Axis-1, Axis-2 and Axis-3. They consciously are not described as X-Y-Z coordinates, because this assignment is determined by the application.

Motor types

2-Phase or 3-Phase stepping motors for bipolar controller technology are supported.

For each motor a separate "D" Type connector is provided.

Motor characteristics

The typical motor characteristics are the holding torque, step angle, maximum phase current, poles and the inductivity. The torque during moving is dependent on the actual speed and has to be taken from the motor characteristic line.

Recommended values:

Phase current	Inductivity	Polepairs	step angle
1,2A - 2.5A	0.8mH - 4mH	25/50/ 100	3.6° / 1.8° / 0.9°

Motor configuration command

The motor adaptation is done via the following Venus-1 commands.

Refer also to the instructions in the Venus-1 Manual.

<i>setumotmin</i>	The motor setting command <code>umotmin</code> affects the phase current when the motor is holding. As a result it determines holding torque and power consumption.
<i>setumotgrad</i>	The motor setting command <code>umotgrad</code> determines the phase current and torque during the motor is moving.
<i>setpolepairs</i>	Adapts the controller to polepairs of the motor. 0.9° Hybridmotor polepairs = 100 1.8° Hybridmotor polepairs = 50

Measures against system resonances

The combination of a motor and other drive components form a typical spring-mass-system.

If the normal damping of the mechanics is insufficient this can lead under inauspicious circumstances to resonance appearances. In the last resort the stepping motor jerks to a stop with a wailing sound.

An effective countermeasure are resonance absorbing elements which are for example installed on the extended motor shaft.



HINWEIS!

For precise positioning tasks we always recommend the use of this damping measure.

A cause for resonances are often overdimensioned or improper motors. In this case it can make sense to reduce the torque via the motor parameter resp. by choosing a more suitable motor.

Motor cable

Upon request Motor cables are produced by us according to customized demands.

If the motor cable is manufactured in-house, cable cross-section, cable length and shielding have to be considered.

The cable cross-section has to be adapted to the phase current and should be calculated based on the maximum phase current per motor phase up to 1500mA.

For cables longer than 4 meters it can be necessary to adapt cable losses by a modified configuration of the motor parameters *setumotmin* and *setumotgrad* or to make up for these losses by an enlarged cable cross-section.

For the EMC-compatible fitting of the cable a total shielding is prescribed whereby the cable shield is connected to the case.

The "D" Type motor connector should have a metal housing. The fasten nuts of the connectors have an UNC4-40 thread.



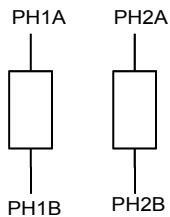
Figure 3: Motor cable shielded with metal housing



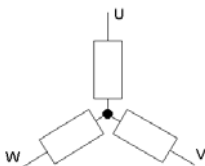
An incorrect wiring of the motor cable can destroy the controller.

Motor wiring

Stepper motor with 2-Phases



Stepper motor with 3-Phases



! It is recommended to use cables that are completely shielded.

The adjustment to the different motor types is accomplished with the following Venus-1 commands:

- ***setphases***
- ***setumotmin***
- ***setumotgrad***
- ***setpolepairs***

Limit Switch Configuration

For each axis there are respectively two Limit Switch inputs available.

The Limit Switch connectors are placed in the respective motor connector. The switch inputs are designated due to their function as cal- Limit Switch Input or as rm- Limit Switch Input. The Limit Switch functions supported by Corvus PCI are linked only to these inputs; no other function can be assigned to them.

The Limit Switch Inputs have an internal filter circuit to filter out any high frequency spikes, which may cause errors in the operation.

Limit Switch Functions

Following Limit Switch Functions are provided:

- Move to the cal-endswitch (Venus-1 command **cal**)

The controller moves to negative direction until the cal-Limit Switch is reached.

- Move to the rm-Limit Switch (Venus-1 command **rm**)

The controller moves to positive direction until the rm-Limit Switch is reached.

Limit Switch Type adaptation

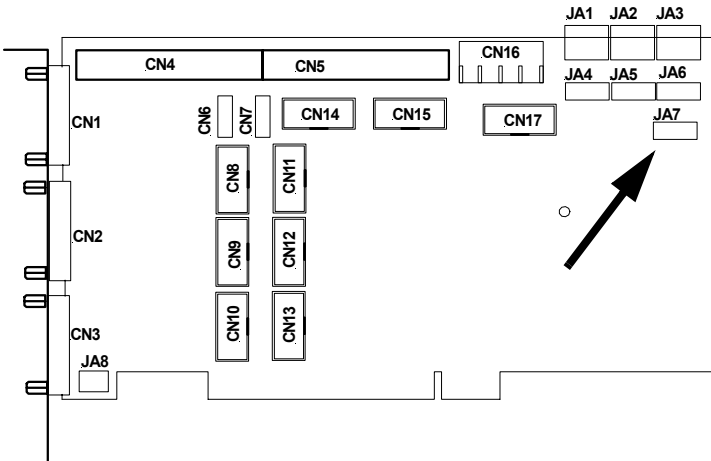
Corvus PCI supports the following types of Limit Switches:

- Mechanical switches (*Opener* or *Closer*)
- Inductive proximity switches NPN or PNP Type
- Photo sensors (NPN/PNP)

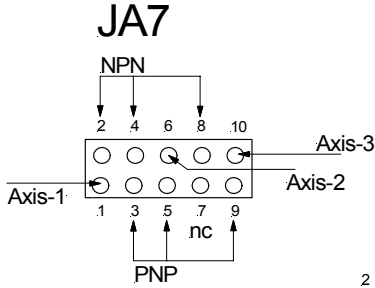
Adaptation to PNP or NPN Types with Jumper Array JA7

- NPN = The Limit Switch closes to Ground or opens
- PNP = The Limit Switch closes to VCC or opens

Type is selected with Jumper Array JA7



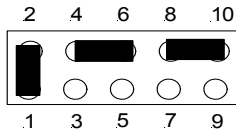
JA7 Function



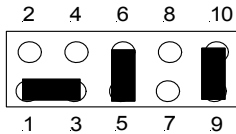
2

JA7 configuration examples:

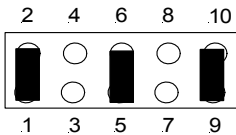
For all axes NPN Type is selected:



For all axes PNP Type is selected:

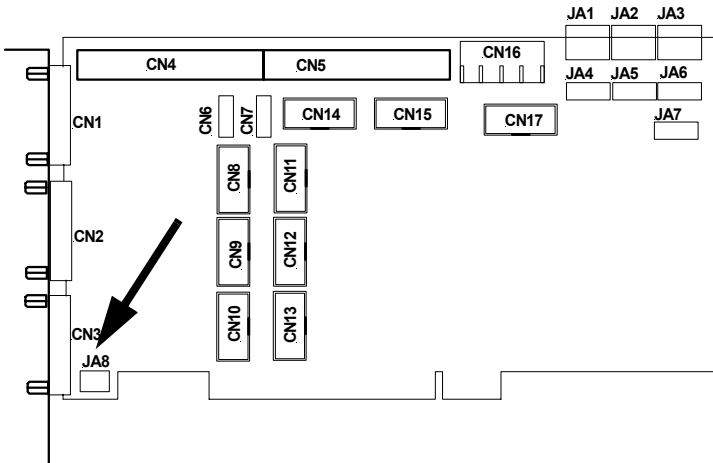


For Axis-1 NPN and Axis-2 and Axis-3 PNP Type is selected

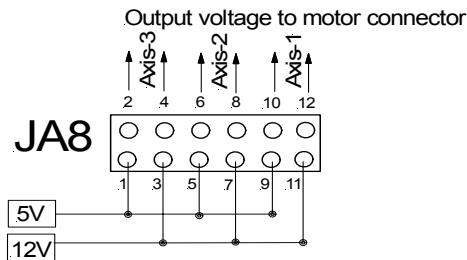


Setting the Switch power supply with Jumper Array JA8

Jumper Array JA8 determines the output voltage value, to supply active Limit Switches with different operation voltages. +5V or +12V can be selected and individually assigned to the motor connectors (Pin8).

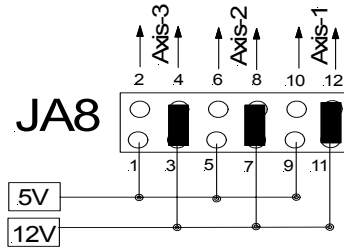


JA8 Function

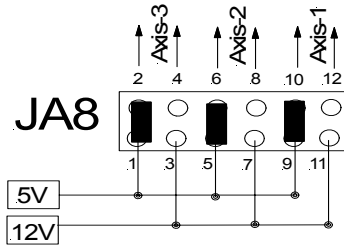


JA8 configuration example

Output voltage 12V, all axes



Output voltage 5V, all axes



Limit Switch adaptation to Opener or Closer Type

- Opener= No active signal means limit has been reached
- Closer = Active signal means limit has been reached

The function *Opener* or *Closer* is determined with the Venus-1 command **setsw**.



Basically, Limit Switches that conclude a movement should be connected as openers. Only in this way can the wire break safety be guaranteed.

Limit Switch wiring

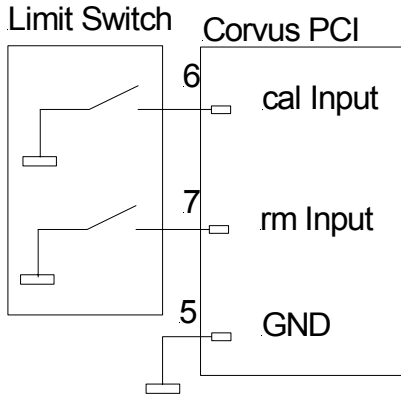


Figure 4: NPN Limit Switch wiring

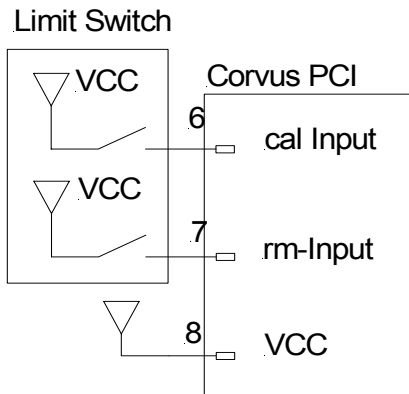


Figure 5: PNP Limit Switch wiring

Chapter 4

Functions

Limit Switch movement

With the Venus-1 commands **cal** and **rm** all active axis are moved synchronous to their Limit Switches. The activation of the axes is done with the command **setaxis**.

Synchronous means the axes are starting at the same time. As soon as all endswitches are activated, the axes are again driven simultaneously out of the endswitches. Therewith the endswitch movement is completed.

The Limit Switch movement can be aborted with the command **Ctrl+c**. During the Limit Switch movement the interpreter is blocked for other commands.



HINWEIS!

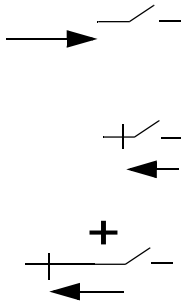
During the endswitch movement no other commands except the abort command **Ctrl+c** can be executed.

The controller can continue to receive data and saves these in the command FIFO. These commands are executed when the Limit Switch movement is completed.

Programmed Limit Switch movement in detail

The fundamental procedure of the programmed Limit Switch movement is described below.

The Limit Switch movement is divided in at least two successive positionings:



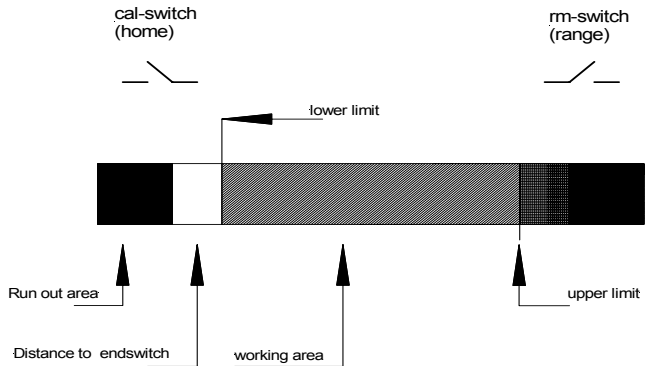
1. The controller positions until the Limit is activated and stops immediately.

2. The controller positions again out of the Switch until it is inactive.

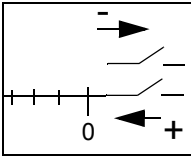
3. With the command **setcalswdist** an additional distance to the endswitches can be determined.

The Limit Switches are always outside the working range. In regular condition the switch will not be activated again.

With the command **getlimit** the registered Limit coordinates are read.

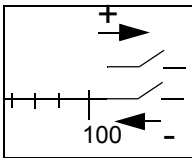


Limit Switch movement to the cal-switch



The controller positions all active axes towards negative position values. At the end of the cal Limit Switch movement the position is set to zero and the origin is firstly determined there. The positioning of coordinates less than zero is no longer possible. The origin can be changed subsequently with the command **setpos**.

Limit Switch movement to the rm-switch



The controller positions all active axes towards positive position values. The position at the end of the rm Limit Switch movement is the maximum movement range for the controller. A positioning beyond this point is no longer possible. The limit can subsequently be minimized with the command **set-limit**. (Software Limit)



For safety reasons it is not possible to save the Limit Switch coordinates. After power up the Limits must be determined again.

Limit Switch movement with the joystick

By the use of the manual positioning with the joystick, the endswitches can also be detected. The procedure resembles the programmed endswitch movement.

Limit Switch movement with the joystick in detail

The controller is positioned with the joystick in the endswitch until this one is activated. After that it is positioned out of the endswitch until this one is again inactive.

The position before the endswitch is registered as the movement limit and can neither in the manual nor in the programmed mode be run over afterwards.

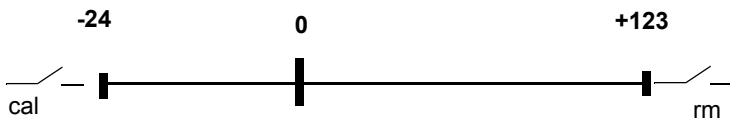
Working range and coordinate system

In opposition to the programmed endswitch movement, the controller does not determine an origin during the manual endswitch movement.

The point of origin could be given here already before the endswitch movement by the command.

If an origin is not determined before the manual endswitch move, the mechanic position at switching on the controller is the origin.

A reassignment of the coordinate origin is subsequently possible.



Notices for the Limit Switch movement



- To calculate the working range and to determine the coordinate system, the origin has to be calculated first (cal). After that the maximum movement range (rm).
- The velocity for the movement to the endswitch has to be chosen that way that within the deceleration distance there will not be a collision with the mechanics. This is first of all valid for the endswitch movement with the joystick.
- To achieve a good reproducibility of the limits, the velocity for the move out of the endswitch should be chosen as small as possible.
- During the endswitch move the command interpreter is blocked for all commands except the abort command Ctrl+c. A
position reply message is only received in terminal mode.
- For safety reasons the movement range and the configurations of the coordinate system are not saved. They have to be recalculated and determined after each restart.

Programmed positioning

Corvus PCI is programmed via the on board PCI to RS-232 bridge with 115200 Baud.

The command language is Venus-1.

The principle of software control is very easy.

There are reading commands and writing commands. Writing commands execute the activities. Reading commands are used to check the controller configuration resp. the current state.

An automatic status reply message can be generated by a combination of commands.

During a positioning task is executed, the status of the controller and actual coordinates can be read.

In the Manual **Venus-1 command language** the possibilities of programming are described in detail.

Venus-1 syntax and examples

The following command groups are distinguished:

- system commands
- configuration commands
- positioning commands
- status commands

System commands

save parameters:

save

restart controller:

reset

Configuration commands

[parameter] _ [axis index] _ [command] _

Example: switching on the 2nd axis

1 2 setaxis

Positioning commands

[axis 1] [axis 2] [axis 3] **move**

Example:

Positioning all three axes to the coordinates below (absolute move)

1.204 4.01 3.4 move

Status commands

System status:

st

Status of the digital inputs:

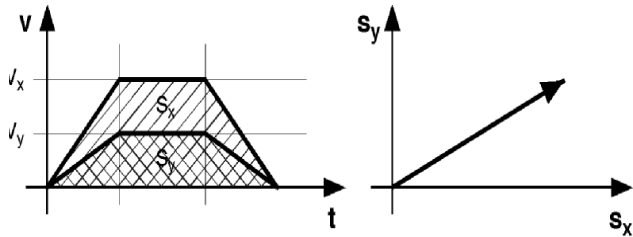
getin

Axes interpolation

In the programmed positioning mode the axes are linear interpolated, i.e., a so-called *vector move* is executed.

In doing so the involved axes are synchronized that way that the phase of acceleration and the phase of deceleration are equal.

A cross table with the axes x and y executes thereby a straight movement.



Velocity and acceleration

The indication of the velocity and the acceleration always refers to the axis with the longest moving distance. It is also called the limiting axis.

The velocity of the other axes v is a result of the ratio of the moving distance of these axes to the moving distance of the limiting axis s_{lim}

$$v = \frac{s}{s_{lim}} \cdot v_{lim}$$

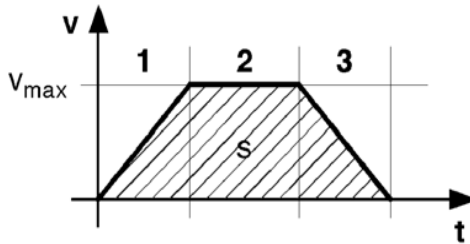
The acceleration of the other axes is a result of their maximum velocities.

Velocity profile of a linear acceleration function

The movement profile of a programmed move which is produced with a linear acceleration function has the form of a trapezium. The area under the curve corresponds to the covered distance s and the pitch of the curve corresponds to the acceleration a .

See v/t graph below.

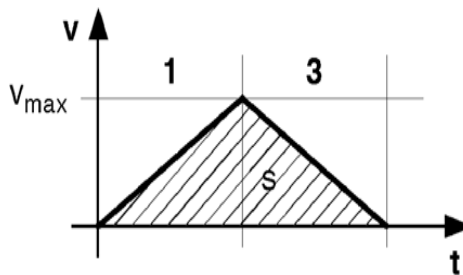
- 1 = accelerated move
- 2 = continuous move
- 3 = decelerated move



Movement profile of short distances

The graph below shows the positioning of a short distance. It can not always be guaranteed here that the axis reaches its maximum velocity.

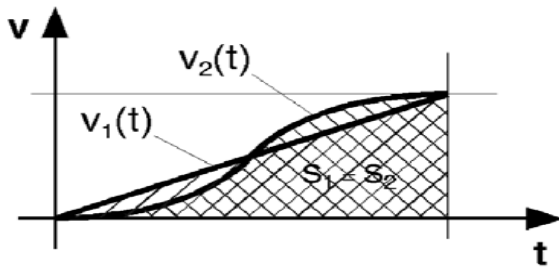
The movement profile has the form of a isosceles triangle.



Movement profile of a \sin^2 -acceleration function

For the \sin^2 -acceleration the torque is not provided abruptly but in form of a sine curve. A very soft start and deceleration of the motor is achieved with it.

As the acceleration is raised in the lower to average speed range it does not lead to an extension of the positioning process.



Closed-Loop Function (Option)

This function enables the operation of the axes in a closed-loop mode.

At this the actual position measured by a measuring system is delivered to the controller to tune the target position exactly.

Depending on the measuring system there are some advantages for the stepping motor positioning:

- sStep monitoring
- Increasing the repeatability
- Improvement of the absolute accuracy
- Load compensation during the move and a standstill
- Improved controlling behavior
- Support of fast and highly precise positioning procedures by the "in-window" function
- Measuring functions

Venus-1 closed loop commands:

- ***setcloop***
- ***setscaleinterface***
- ***setclperiode***
- ***setclfactor***
- ***setscaletype***
- ***setclpara***
- ***setref***
- ***refmove***

Supported measuring systems

Corvus supports the following measuring systems:

- A/B Quadrature with RS422 signal output
- SIN/COS, 1Vss

A/B Quadrature input



The interface supports square wave pulse trains and their inverted pulse trains.

The encoder interface is also provided with a reference signal input.

Technical data:

Incremental signal input	RS422
Encoder index input	RS422
Max. input frequency	20 Mhz
Max. input resolution	0.233 nm (233fm)

SIN/COS Module (Option)

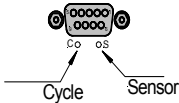


The SIN/COS module is an independent data interface for sinusoidal encoder signals. The encoder signals are interpolated with a 12 Bit or 16Bit resolution. For each axis a SIN/COS module is required.

Technical data:

Input signals	1Vss, differential
Reference mark	Supported
Max. input frequency	500,000 sine periods/second
Interpolation	12 Bit or 16Bit

SIN/COS Status LED



Cycle LED	Status
active	Hardware OK
not active	Hardware error

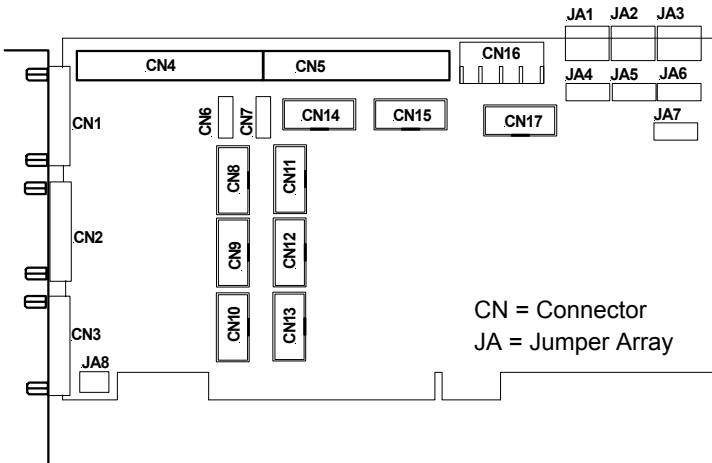
Sensor LED	Status
active	no Sensor
not active	Sensor OK

.....

Chapter 5

Electrical Interface

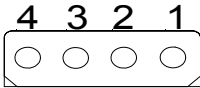
Electrical Interface Locations and Designation



- CN1: Motor and Limit Switch Axis-1
- CN2: Motor and Limit Switch Axis-2
- CN3: Motor and Limit Switch Axis-3
- CN4: Sin-Cos Encoder Input
- CN5: LED Status Display, Analog Out
- CN6: CAN-Bus
- CN7: Debug Interface COM0
- CN8: Digital Encoder Input Axis-2
- CN9: Digital Encoder Input Axis-3
- CN10: Digital Encoder Input Axis-1
- CN11: Joystick Interface
- CN12: Digital I/O Interface
- CN13: Safety Device Interface
- CN14: Factory use
- CN15: Factory use
- CN16: Molex Power Connector
- CN17: Factory use

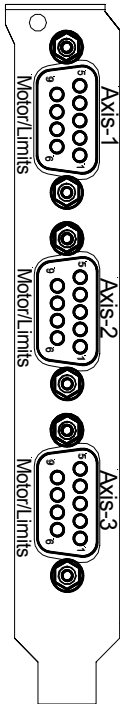
- JA1: Factory use
- JA2: External Power select
- JA3: Factory use
- JA4: Factory use
- JA5: RS-232 Baudrate select
- JA6: Factory use
- JA7: NPN/PNP Limit Switch type
- JA8: 5/12V Limit Switch voltage

Power connector CN16



Pin	Name	Description
1	+12V	Power Motor Driver
2	GND	Ground
3	GND	Ground
4	+5V	Power Controller

Motor and Limit Switch Connectors CN1-CN3



Pin	Name	Description
1	Ph 1A / U	Motor phase 1A
2	Ph 1B / W	Motor phase 1B
3	Ph 2A / V	Motor phase 2A
4	Ph 2B	Motor phase 2B
5	GND	
6	cal*	Negative Limit Switch Input
7	rm*	Positive Limit Switch Input
8	5V/12V	Limit Switch voltage
9	Temp	Temp. Sensor Input